WHAT IS CLAIMED IS:

1. A computer-implemented programming method, comprising:

providing a runtime environment including a client application implementing an object model, the object model including a hierarchy of object classes capable of being instantiated in the client application, the object model also including one or more class templates, each class template having an associated class type and including class functionality information for a custom class capable of being implemented in the client application, the class functionality information for each class template including a set of required class functionality for the associated class type and a set of elective class functionality for the associated class type;

receiving in the client application a first user input including a custom class creation request specifying a class type for a custom class;

in response to the first user input, launching a design-time environment for defining custom object classes;

retrieving the class functionality information for the class template associated with the specified class type;

receiving in the design-time environment a second user input specifying a subset of the elective class functionality to be implemented in the custom class; and

generating in the design-time environment a class definition defining a custom class having the required class functionality and the specified elective class functionality, wherein objects belonging to the custom class are capable of being instantiated from the runtime environment independent of the design-time environment.

2. The method of claim 1, wherein:

the class functionality information includes information identifying a set of mandatory interfaces defining the required class functionality for the associated class type and information identifying a set of elective interfaces defining the elective class functionality for the associated class type.

3. The method of claim 2, further comprising:

after retrieving the class functionality information for the class template associated with the specified class type, displaying to the user a list of the set of elective interfaces defining the elective class functionality for the associated class type; and

wherein receiving the second user input includes receiving a user selection of one or more of the elective interfaces from the displayed list of elective interfaces.

4. The method of claim 3, wherein:

generating the class definition includes aggregating a plurality of objects implementing the mandatory interfaces and the selected elective interfaces.

5. The method of claim 4, wherein each of the elective interfaces has a set of associated interface properties and interface methods, the method further comprising:

receiving in the design-time environment a third user input including one or more code fragments further defining one or more of the interface properties and/or interface methods associated with at least one of the selected elective interfaces.

6. The method of claim 5, further comprising:

generating code skeletons for one or more of the interface properties and interface methods associated with the selected elective interfaces, each code skeleton being derived from object metadata describing the specified elective interface that is associated with the interface property or interface method; and

displaying the code skeletons to the user.

7. The method of claim 1, wherein:

the design-time environment includes a graphical user interface; and

the runtime environment is independent of the graphical user interface.

8. The method of claim 1, further comprising:

instantiating an object of the custom class in the runtime environment.

9. The method of claim 8, wherein:

the class definition is generated as a dynamic linked library; and

instantiating the object includes calling the dynamic linked library in the runtime environment.

10. The method of claim 9, wherein:

the runtime environment includes a wrapper object operable to execute code in the dynamic linked library.

11. The method of claim 1, wherein:

the client application is an automated process control program and the custom class is a resource class for defining an apparatus driver for an apparatus coupled to the automated process control program.

12. The method of claim 1, wherein:

the client application is an automated process control program and the custom class is an action class for defining an action capable of being implemented by the automated process control program.

13. A computer program product on a computer-readable medium for generating custom object classes, the program comprising instructions operable to cause a programmable processor to:

provide a runtime environment including a client application implementing an object model, the object model including a hierarchy of object classes capable of being instantiated in the client application, the object model also including one or more class templates, each class template having an associated class type and including class functionality information for a custom class capable of being implemented in the client application, the class functionality information for each class template including a set of required class functionality for the associated class type and a set of elective class functionality for the associated class type;

receive in the client application a first user input including a custom class creation request specifying a class type for a custom class;

in response to the first user input, launch a design-time environment for defining custom object classes;

retrieve the class functionality information for the class template associated with the specified class type;

receive in the design-time environment a second user input specifying a subset of the elective class functionality to be implemented in the custom class; and

generate in the design-time environment a class definition defining a custom class having the required class functionality and the specified elective class functionality, wherein objects belonging to the custom class are capable of being instantiated from the runtime environment independent of the design-time environment.

14. The computer program product of claim 13, wherein:

the class functionality information includes information identifying a set of mandatory interfaces defining the required class functionality for the associated class type and information identifying a set of elective interfaces defining the elective class functionality for the associated class type.

15. The computer program product of claim 14, further comprising instructions operable to cause a programmable processor to:

after retrieving the class functionality information for the class template associated with the specified class type, display to the user a list of the set of elective interfaces defining the elective class functionality for the associated class type; and

wherein the instructions to receive the second user input include instructions operable to cause the programmable processor to receive a user selection of one or more of the elective interfaces from the displayed list of elective interfaces.

16. The computer program product of claim 15, wherein:

the instructions to generate the class definition include instructions operable to cause a programmable processor to aggregate a plurality of objects implementing the mandatory interfaces and the selected elective interfaces.

17. The computer program product of claim 16, wherein each of the elective interfaces has a set of associated interface properties and interface methods, the computer program product further comprising instructions operable to cause a programmable processor to:

receive in the design-time environment a third user input including one or more code fragments further defining one or more of the interface properties and/or interface methods associated with at least one of the selected elective interfaces.

18. The computer program product of claim 17, further comprising instructions operable to cause a programmable processor to:

generate code skeletons for one or more of the interface properties and interface methods associated with the selected elective interfaces, each code skeleton being derived from object metadata describing the specified elective interface that is associated with the interface property or interface method; and

display the code skeletons to the user.

19. The computer program product of claim 13, wherein:

the design-time environment includes a graphical user interface; and

the runtime environment is independent of the graphical user interface.

20. The computer program product of claim 13, further comprising instructions operable to cause a programmable processor to:

instantiate an object of the custom class in the runtime environment.

21. The computer program product of claim 20, wherein:

the class definition is generated as a dynamic linked library; and

the instructions to instantiate the object include instructions operable to cause the programmable processor to call the dynamic linked library in the runtime environment.

22. The computer program product of claim 21, wherein:

the runtime environment includes a wrapper object operable to execute code in the dynamic linked library.

23. The computer program product of claim 13, wherein:

the client application is an automated process control program and the custom class is a resource class for defining an apparatus driver for an apparatus coupled to the automated process control program.

24. The computer program product of claim 13, wherein:

the client application is an automated process control program and the custom class is an action class for defining an action capable of being implemented by the automated process control program.

25. A computer-implemented custom class design system, comprising:

means for providing a runtime environment including a client application implementing an object model, the object model including a hierarchy of object classes capable of being instantiated in the client application, the object model also including one or more class templates, each class template having an associated class type and including class functionality information for a custom class capable of being implemented in the client application, the class functionality information for each class template including a set of required class functionality for the associated class type and a set of elective class functionality for the associated class type;

means for providing, in response to a first user input including a custom class creation request specifying a class type for a custom class, a design-time environment for defining custom object classes;

means for retrieving the class functionality information for the class template associated with the specified class type and receiving in the design-time environment a second user input specifying a subset of the elective class functionality to be implemented in the custom class; and

means for generating in the design-time environment a class definition defining a custom class having the required class functionality and the specified elective class functionality, wherein objects belonging to the custom class are capable of being instantiated from the runtime environment independent of the design-time environment.